18.338 Eigenvalues of Random Matrices

Problem Set 3

Due Date: Mon Oct. 19th, 2020

Reading and Notes

Read chapter *7* of the class notes. notes. Please again give your feedback especially high level style and where things did not make sense, in addition to spelling or technical errors

Homework

Do the first 3 exercises plus two out of the rest 5.

1. (M) The Christoffel-Darboux formula (also see equation (5.6) on page 78) states

$$\sum_{j=0}^{n} \pi_i(x)\pi_j(y) = \frac{k_n}{k_{n+1}} \frac{\begin{vmatrix} \pi_n(x) & \pi_n(y) \\ \pi_{n+1}(x) & \pi_{n+1}(y) \end{vmatrix}}{y-x},$$

   where k_n is the lead coefficient of π_n. Let

$$\pi_j(x) = \frac{H_j(x)}{(\sqrt{\pi} j! 2^j)^{1/2}}$$

   for Hermite Polynomials, then $k_n/k_{n+1} = \sqrt{n}$.

   Use the known asymptotics

$$\lim_{m\to\infty} (-1)^m m^{1/4}\ \pi_{2m}(x)e^{-x^2/2} = \frac{\cos(\xi)}{\sqrt{\pi}}$$
$$\lim_{m\to\infty} (-1)^m m^{1/4}\ \pi_{2m+1}(x)e^{-x^2/2} = \frac{\sin(\xi)}{\sqrt{\pi}}$$

   where $x = \xi/(2\sqrt{m})$ to prove

$$K_{2m}(x,y) = e^{-(x^2+y^2)} \sum_{j=0}^{2m-1} \pi_j(x)\pi_j(y) \tag{1}$$

   converges to the sine kernel

$$2\sqrt{\pi} m \,\frac{\sin(x-y)}{x-y}.$$

2. (C) Do a numerical experiment to "see" the convergence in Problem 1. There are numerical issues on the diagonal and corners. Probably on the diagonal, Christoffel-Darboux needs to be replaced by a derivative approximation. See if you can make it better.

3. (C) Obtain the Airy Process limit by taking numerically

$$\frac{1}{\sqrt{2}\, n^{1/6}} K_n(\sqrt{2n} + \frac{x}{\sqrt{2}\, n^{1/6}}, \sqrt{2n} + \frac{y}{\sqrt{2}\, n^{1/6}}) \to \frac{Ai(x)Ai'(y) - Ai'(x)Ai(y)}{x-y},$$

   where $Ai(x)$ is the Airy function and K_n is defined in (1).

4. (C) On paper or by computer see if you can implement lanczos on the Hermite weight function,say, to derive the Hermite tridiagonal. By computer consider the process described here https://www.mathworks.com/examples/matlab/community/12733-orthogonal-polynomials-via-the-lanczos-process, but use Julia.

   We recommend using the Approx Fun Package (https://github.com/ApproxFun/ApproxFun.jl, and in particular the Lanczos routine in https://github.com/JuliaApproximation/ApproxFun.jl/blob/master/src/Extras/lanczos.jl), and the wonderful blog post by Sheehan Olver (https://approximatelyfunctioning.blogspot.com/2020/09/quasi-matrices-orthogonal-polynomials.html)

   Also one could try doing it exactly in Mathematica. Mathematica's success or failure will depend on the ability to do integrals.

5. (C) It would be of interest to see if one can use similar methods in Julia to evaluate the matrix kernels, say the sine kernel or the airy kernel. Plot these.

6. (M) Exercise 7.1 (p132)

7. (M) Exercise 7.2 (p132)

8. (M) Exercise 7.7 (p133)